UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/649,270	08/28/2000	Lawrence A. Crowl	SUN1P380/P4501	6759

22434 7590 02/26/2004

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

EXAMINER	
VU, TUAN A	

ART UNIT	PAPER NUMBER
2124	8

DATE MAILED: 02/26/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PPG

Office Action Summary	Application No. 09/649,270	Applicant(s) CROWL ET AL.	
	Examiner Tuan A Vu	Art Unit 2124	

-- Th MAILING DATE of this communication appears on the cover she t with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 January 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-16 and 18-21* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-16 and 18-21* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 1/12/2004.

As indicated in Applicant's response, claims 1, 3-6, 10-13, 16, 19, and 21 have been amended. Claims 1, 3-16, and 18-21 are pending in the office action.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 7 recites the limitation "the source program" in line 1. There is insufficient antecedent basis for this limitation in the claim. To examine the merits of the claim, this limitation will be understood as though it were compiler information or data.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 7-8, 10-13, 16, and 21 are rejected under 35 U.S.C. 103(a) as being unpatentable over Unger et al., USPN: 5,991,173 (hereinafter Unger), in view of Porter, USPN: 6,163, 811 (hereinafter Porter).

As per claim 1, Unger discloses a method of generating compiler products (e.g. Fig. 7) in a compressed form, said method comprising:

compressing a portion of compiler information to obtain compressed compiler information (e.g. steps *210, 212, 213, 214*, Fig. 8); and

producing a compressed compiler product based on at least the compressed compiler related information (e.g. step 218, 219, Fig. 8; col. 12, lines 1-6).

But Unger does not explicitly specify the portion of the compiler information being compressed comprises encoded program symbol names. Unger however discloses that the compiler information being compressed includes textual symbol names (e.g. *vocabulary word, token* – col. 9, lines 5-30; *numeric string, currency symbols* – col. 10, lines 23-39) and using some compression encoding scheme (e.g. *dictionary ... encoding, run-length encoding* - col. 10, line 23 to col. 11, line 44); hence has implicitly disclosed some form of encoding the textual and numerical symbols as well as implicitly disclosed a form of encoding text strings under some format required for HTML protocol (e.g. col. 4, lines 25-55). The use of link and hyperlinks in HTML documents and web pages like in Unger's method to embed programming language constructs (e.g. Javascript, Java code statements) and scripting constructs was a well-known concept at the time the invention was made. Porter, in a method to tokenize source code for compression like Unger, discloses parsing tokenized programming constructs, i.e. symbols (e.g. Fig. 1a-c, 2a-b, 3a-b), hence teaches applicability of processing scheme onto browser textual constructs (e.g. Fig. 4; col. 2, line 21-34) as well as programming language constructs in conjunction with browser technologies as suggested by Unger. In case Unger's method does not already compress encoded program language constructs or symbols, it would have been obvious

for one of ordinary skill in the art at the time the invention was made to implement the parsing of text or HTML documents by Unger so as to further include the parsing of encoded programming language symbols as taught by Porter because natural language and encoded program symbols are closely related for being human readable language; and the effect of such parsing ability would further extend the method of compressing and improve product marketability (browser and programming source code) for Unger.

As per claim 7, Unger discloses that the source program to compile is HTML material such as HTML, XML, SGML files (e.g. col. 5, lines 1-12); but does not specify that the source program is a programming language written in C++, Java, Pascal, or Fortran. Porter, in a method to compress application code using tokenized source data and symbol storage and web page for source file (e.g. col. 2, line 21-34) as mentioned above, discloses applying compression to Java program source code (e.g. col. 4, lines 6-24; Figs. 1-3). It would have been obvious for one of ordinary skill in the art at the time the invention was made to use a source program written in Java as taught by Porter and submit it to the compression process used by Unger because Java language programming and its products are well-known for their portability and platform independency as well as support of many browser applications and material, i.e. HTML, XML applications just as suggested in Unger's invention.

As per claim 8, Unger discloses parsing and compressing browser documents but Porter from above discloses compressing of program code using tokenized process analogous to Unger. In view of the rationale in claim 1 using Porter's teachings for addressing the program code symbols parsing, the limitation as to compress an object code file would also have been obvious herein because one ordinary skill in the art would be motivated to combine using the browser

compiler/parsing schemes by Unger and enhance those with capabilities to parse program code and compress such code as taught by Porter in order to yield compressed version of such parsed program as intended by Unger, because object code delivery in compressed form would facilitate distribution and storage resources saving.

As per claim 10, Unger discloses a method for generating uncompressed symbol names being associated (col. 9, lines 5-30; col. 10, lines 23-39) with compiler information, said method comprising:

identifying a compressed encoded symbol name being associated with compiler information (*token, words, strings* – col. 16, lines 8-17; Fig. 5 – Note: token is compiler information and symbol formatted inside a HTTP protocol is equivalent to being encoded under such HTML format or HTTP protocol);

obtaining information relating to the compressed symbol name (e.g. *dictionaries* – col. 38-55); and

decompressing the compressed encoded symbol name to obtain an encoded symbol name in a uncompressed form (e.g. col. 15, line 60 to col. 16, line 7).

But Unger does not specify that the uncompressed or compressed encoded symbol names are encoded program symbol names. But this encoded program symbol name limitation would have been obvious in view the corresponding rejection set forth in claim 1 using Porter's teachings, and is rejected herein with the same rationale as set forth therein.

As per claim 11, Unger suggests obtaining information (*tag tree 54, vocabulary 58* – Fig. 7) referenced by an encoded program symbol reference (*hyperlinks* – col. 7, lines 32-44; *tag* -- Fig. 4,5) included in the compressed encoded program symbol name (e.g. step *214, 220,*

compressed objects/text - Fig. 8), such symbol reference providing a reference to a base encoded program symbol (*files A and J, pages C, I* -- col. 7, lines 32-44; Fig. 5) that is associated with the encoded program symbol name represented by the compressed encoded program symbol name (*compressed objects/text* – Fig .8; col. 9, lines 5-30; col. 10, lines 23-39). Further, Porter teaches pointer referencing symbol table (e.g. Fig. 2a-b); hence, it would have been obvious for one of ordinary skill in the art at the time the invention was made to modify the referencing as taught by Unger for browser documents so as to implement the pointing technique as taught by Porter when applying to programming language because this would enhance the compression method by Unger in case this method would extend itself in tokenizing and compressing programming language for the benefits as set forth in claim 1 above.

As per claim 12, Unger (with Porter's teachings) further discloses that the program base symbol is the container of the symbol represented by the compressed encoded program symbol name (e.g. *files A and J, pages C, I* -- col. 7, lines 32-44 – Note: compressed text or numerals in objects of Fig. 8 are contained in files or pages referenced by links, tag or hyperlinks in Fig. 5. This is equivalent to base symbol, or containers, e.g. files/pages A, J, C, I, for text data symbols compressed in objects/text of Fig. 8; and HTML text token is equivalent to encoded symbols for HTTP protocol and HTML format).

As per claim 13, Unger discloses a compilation system suitable for compiling source programs, said compilation system comprising:

an enhanced compiler suitable for generation of enhanced compiler products (products *54-62* – Fig. 7), such compiler compiles a source program to produce enhanced compiler

products with a reduced size in comparison with conventional compiler products (e.g. col. 1, line 47 to col. 2, line 39; steps *210, 212, 213, 218* -- Fig. 8); and

at least one enhanced non-compiler component that understands and utilizes the enhanced compiler products (e.g. *proxy* – col. 14, lines 14-58).

But Unger does not explicitly specify that the source program has at least one compressed encoded program symbol name. But this encoded program symbol name limitation has been addressed in the corresponding rejection set forth in claim 1 using Porter's teachings.

As per claim 16, this is a computer-readable medium claim corresponding to claim 1 above, including all the limitations therein, hence is rejected herein for the same reasons as set forth therein.

As per claim 21, this is a computer-readable medium claim corresponding to claim 10 above, including all the limitations therein, hence is rejected herein for the same reasons as set forth therein.

6. Claims 3-6, 18, and 19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Unger et al, USPN: 5,991,173, and Porter et al., USPN: 6,163, 811, as applied to claims 1, 16 above, in view of Mogul et al., "Potential benefits of delta encoding and data compression for HTTP", Sept. 1997(hereinafter Mogul), and further in view of Schaumont et al., USPN: 6,606,588 (hereinafter Schaumont).

As per claim 3, Unger (with Porter's teachings) discloses compressing operable to reduce the length of a plurality of encoded program symbol names using encoding schemes (*Huffman, Lempel-Ziv* -- col. 8. lines 45-52; *run-length encoding* – col. 11, col. 38-44); but does not specify such encoding scheme is differential encoding scheme. One of ordinary skill in the

art would recognize that the differential encoding scheme is analogous to the delta encoding which encodes the difference between 2 sets of data. Besides, the learning on the entropy of the data leading to the use of Huffman, Lempel or run-length and differential encoding of the data in conjunction with the application of these encoding schemes as to complement each other according to the level of entropy observed was a well-known concept in the art of compression of data for storage or transmission. One of ordinary skill in the art would recognize Huffman's or Lempel-Ziv's encoding (using predictive and probabilistic computation, respectively) to be complementary to differential encoding scheme inasmuch as their similar approach in taking into account data arrangement/distribution prediction and entropy change. Such differential encoding scheme, or delta encoding, reduces storage as well as demands less computation in regard to the subsequent decoding or reconstructing of encoded data transmitted across the internet. Likewise, Mogul, in a system to reduce memory resources in the transmission of HTTP data analogous to the data size reduction technique used by Unger's compression method, discloses using delta encoding (i.e. differential encoding) of HTTP message and packet streams (e.g. ch. 4, 5 – pg. 3-10). Further, in a system using a compiler to compact a C++ program-based behavioral model and its metadata to facilitate transmission in a network associated with an object-oriented programming and MATHLAB framework, Schaumont also discloses a differential encoding scheme (e.g. *differential encoder* - col. 71, line 26 to col. 74, line 47; Fig. 23, 24). It would have been obvious for one of ordinary skill in the art at the time the invention was made to implement the differential encoding scheme as taught by Mogul for HTTP type content transmission and by Schaumont for object-oriented language constructs transmission to further complement the encoding techniques mentioned by Unger (with Porter's teachings) because the

differential scheme uses a simpler algorithm to complement to Unger's encoding approach, i.e. predictive computation related resources used in Huffman or Lempel-Ziv encoding as suggested by Unger, thus improving computing resources usage efficiency, such efficiency amounting to a less complicated decoding/reconstructing scheme as taught by Mogul and Schaumont from above.

As per claim 4, Unger (with Porter's teachings) discloses encoding an encoded program symbol name in the compiler information (Fig. 8; col. 8. lines 45-52) with an encoded format but does not explicitly specify identifying an encoded program symbol name that is encoded in an extended format encoding; nor determining a differential encoding for the encoded symbol name; nor replacing the extended format encoding for the symbol name with the differential encoding. The limitation as to compress encoded program symbols (re claim 1: Unger's encoding scheme and encoding under HTML format) has been addressed in claim 1 above. Further, in view of the teachings by Mogul and Schaumont to encode symbol characters or integers using the delta, just as mentioned in Mogul and Schaumont' system in claim 3 above, one of ordinary skill in the art would recognize therein the presence of both symbols prior to the delta being computed and those making up the delta list, hence it would have been obvious for one of ordinary skill in the art at the time the invention was made to provide the differential encoding by Mogul and Schaumont and apply it to Unger's method and establish the distinction between the encoded symbols prior to the delta extraction, i.e. identifying the extended format encoded symbols, and the symbols making up the delta portion, i.e. determining the differential encoded (delta) symbols, in order to replace the extended format of such symbols by the

differential encoded format thereof. One of ordinary skill in the art would be motivated to do so because of the same reasons rendering claim 3 obvious as set forth therein.

As per claim 5, Unger (with Porter's teachings) discloses determining an encoded program symbol name identifier (e.g. *token* – col. 8, line 54 to col. 9, line 14; *token range* – Fig. 9); and attaching such identifier to the encoding (Fig. 8; *token numbers* -- col .9, lines 39-54; steps *210,212* – Fig. 8). However, Unger does not disclose that such encoding is a differential encoding; but this limitation has been addressed for obviousness in claim 3 above and herein is rejected for the same rationale therein.

As per claim 6, Unger discloses a container reference to indicate a container name associated with at least one of the encoded program symbol names (e.g. *Token Range* – Figs.9, 10) and Porter disclose a class object as object representing a container referencing to a container name associated with encoded symbol names declared inside the object or class (e.g. *pointer* - Fig. 2a, 2b). The motivation to combine Unger teachings and Porter teachings to provide the name identifier equivalent to pointer to a class object associated with program symbols is the same as that set forth in claim 1.

As per claim 18, this is a computer-readable medium version of claim 3 above; and includes all the limitations of claim 3; hence is rejected herein for the same reasons set forth therein.

As per claim 19, this is a computer-readable medium version of claim 4 above; and includes all the limitations of claim 4; hence is rejected herein for the same reasons set forth therein.

7. Claims 9, 15 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Unger et al, USPN: 5,991,173, and Porter et al., USPN: 6,163, 811, as applied to claims 1, 13, 16 above, in view of (no author) G06F011/28 by Derwent 1998-236084, JP Pub N: JP 10074152A (hereinafter JP-DW-1998)

As per claim 9, Unger only discloses that the compressed compiler related product has some debugger information (e.g. *tag tree* -- Fig. 7; *determining how* -- col. 7, lines 39-64; col. 14, lines 18-54; Fig. 13; step *700*, *logfile 702* – Fig.14) via the tag tree and dictionary which are information used in conjunction with consistency checking between hyperlinks and related compressed HTML files, i.e. debug information supportive of the hypertext material re-assembling/binding from transfer across a network. But Unger does not specify compressing such debug information into Unger's compiler related product. JP-DW-1998, in a debug system for compressing and delivering compressed program code similar to Porter's method, discloses including debug information as suggested by Unger in the compressed form suitable for storage (JP-DW-1998: see front page and abstract). Hence, It would have been obvious for one of ordinary skill in the art at the time the invention was made to include the information used by Unger (with Porter's enhancements) to help debug the compressing/parsing process as disclosed in the compressed compiler product as suggested by JP-DW-1998 because such debug material would enable debug and/or support the execution of the code when uncompressed and utilized by the recipient to which the code is delivered.

As per claim 15, Unger (with Porter's teachings) teaches an enhanced compressed compiler product selected from being an object file, but fail to specify including therein an executable and a debugging information. JP-DW-1998, in a debug system for compressing code

as mentioned in claim 9 above, discloses compressing both the debug information and executable code in the deliverable that is to be loaded on the target computer (JP-DW-1998: see front page and abstract). In view of the rationale in claim 8 to combine Unger teachings with Porter's for providing object file code, it would also be obvious for one of ordinary skill in the art at the time the invention was made to further include executable code and debug information in the compressed product as taught by JP-DW-1998 in order to enhance the utilization of the compressed code delivered as suggested by Porter as to facilitate the debugging and additional memory usage as suggested by JP-DW-1998.

As per claim 20, this claim includes the same limitations as claim 16 above, hence is rejected herein for the same reasons as set forth therein.

8. Claim 14 is rejected under 35 U.S.C. 103(a) as being unpatentable over Unger et al, USPN: 5,991,173, and Porter et al., USPN: 6,163, 811, as applied to claim 13 above, in view of Klein S.T., Bookstein A., Deerwester S., "Storing Text Retrieval Systems on CD-ROM: Compression and Encryption Considerations", July 1989, ACM Trans. *On Information Systems 7*, pp. 230–245(hereinafter Klein).

As per claim 14, Unger (with Porter's teachings) discloses using encoding technique to reduce size of the enhanced compiler product (Unger: *Huffman* -- col. 8. lines 45-52), e.g. Huffman encoding; but does not specify that such reduction is up to 40 percent of sizes of conventional compiler products. Klein, in a analogous method to compress text data for a storage medium, discloses that Huffman encoding can achieve between 48% and 52.5% compression of English text, using different character bytes setting (p. 5, last paragraph) or between 40-65 % for text or dictionary, respectively (p. 22, Table 1). It would have been

obvious for one of ordinary skill in the art at the time the invention was made to implement the statistical results by Huffman encoding such as taught by Klein into Unger's technique of compressing using the same encoding technique because targeting and achieving up to 40% in size reduction would better preserve storage resources as intended in Unger's compression technique.

Response to Arguments

9. Applicant's arguments filed 9/22/2003 have been fully considered. Following are the Examiner's observations in regard thereto.

(A) As per claims1 and 16, Applicants have submitted that Unger and Porter 'leave encoded program symbol names unaltered and uncompressed', and that Unger does not compress encoded program symbol names and that Porter describes 'only compression of entire source files over distributed network' (Appl. Rmrks, pg. 9, 1st para). The claim recites 'compressing comprises encoded program symbol names' and 'compressing a portion of compiler information'. As interpreted, the claimed features do not enforce how a particular or specific format of program symbol name is encoded by compression while excluding the rest of a document symbols. The rejection has pointed out that Unger uses some scheme to compress textual content of markup files, and by submitting textual data under some markup process, some level of special encoding is implicitly disclosed. The rejection has also shown that if Unger does not already enclose program symbols (as in a programming code symbols) inside a marked up content, it would have been obvious to combine the compression of Java source files by Porter to Unger's compressing process by virtue of the motivation as set forth therein. It is well recognized that rendering a HTML, SGML or XML document is also a form of encoding being

applied to textual tokens or symbols to conform those respective standards. Thus, creating specially formatted documents like Java source files or markup pages already encompasses a certain level of encoding of textual tokens inside some formatting scheme or algorithm according to the language format and grammar. The claim does not specify how the limitation recited as 'compressing comprises encoded program symbol names' clearly distinguishes over the process of compressing encoded program symbols by either Unger or Porter. Indeed, a source file by Porter or a markup page with hyperlinks by Unger can very much comprise encoded program symbol names in view of the implied encoding of such symbols while creating such files/pages as has been pointed out above. Further, the recited 'compressed compiler related information' does not spell out that only some form of symbols are encoded while leaving out some other forms of textual content. For the sake of argument, even if such encoding during the formation of Unger's markup files is not considered a encoding of symbols, the fact of compressing text tokens by a compressing algorithm (e.g. Huffman, dictionary encoding) already encompasses a form of encoding data in accordance to such particular compressing algorithm, hence does read on 'compressing of encoded program symbols'.

(B) As per claims 3-6, 18, 19, and the arguments concerning Burrows in relation to with the differential encoding scheme (Appl. Rmrks, pg. 9, bottom, pg. 10, top), the arguments are now moot in view of the current grounds of rejection.

Conclusion

10. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat No, 4,942,467 Waldman et al., disclosing differential/predictive encoding to symbols transmitted prior to demodulation.

U.S. Pat No. 6,442,680 to Elnozahy, disclosing entropy-based encoding for RISC instructions for compression.

Applicant's arguments filed 9/22/2003 have been fully considered but they are moot in view of new grounds of rejection.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (703)305-7207. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(703) 872-9306 (for formal communications intended for entry)

or: (703) 746-8734 (for informal or draft communications, please label "PROPOSED" or "DRAFT" – please consult Examiner before use)

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington. VA. , 22202. 4th Floor(Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

VAT
February 22, 2004

KAKALI CHAKI
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100